

WPROWADZENIE DO

# HTML

## SPRAWDŹ, CO POTRAFI HTML5!

- Jakie nowości wprowadza HTML w wersji 5?
- Jak wykorzystać usługi geolokalizacyjne?
- Jak tworzyć dostępne witryny WWW?

# 5

**AUTORYTETY INFORMATYKI**

BRUCE LAWSON, REMY SHARP



## » Idź do

- Spis treści
- Przykładowy rozdział
- Skorowidz

## » Katalog książek

- Katalog online
- Zamów drukowany katalog

## » Twój koszyk

- Dodaj do koszyka

## » Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

## » Czytelnia

- Fragmenty książek online

## » Kontakt

Helion SA  
ul. Kościuszki 1c  
44-100 Gliwice  
tel. 32 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)  
© Helion 1991–2011

## Wprowadzenie do HTML5. Autorytety Informatyki

Autor: Bruce Lawson, Remy Sharp

Tłumaczenie: Radosław Meryk

ISBN: 978-83-246-3045-5

Tytuł oryginału: [Introducing HTML5 \(Voices That Matter\)](#)

Format: 168×237, stron: 240



### Sprawdź, co potrafi HTML5!

- Jakie nowości wprowadza HTML w wersji 5?
- Jak wykorzystać usługi geolokalizacyjne?
- Jak tworzyć dostępne witryny WWW?

**HTML5** – początkowo po cichu, ostatnio coraz głośniejsze – zaczyna gościć w naszych przeglądarkach internetowych. Wbudowane mechanizmy obsługi multimediów, lepsze wykorzystanie formularzy, nowe struktury i semantyka, dzięki którym Twój serwis stanie się bardziej dostępny, to tylko część z nowinek ostatniej wersji HTML. Zastanawiasz się, jak wykorzystać go na Twoich stronach?

Ta książka odpowie na to i wiele innych pytań! Kompletnie źródło informacji na temat HTML5 jest w zasięgu Twojej ręki. W trakcie lektury poznasz nowe struktury i ich przeznaczenie oraz zmodyfikowane znaczniki z wcześniejszych wersji. W kolejnym rozdziale zaznajomisz się szczegółowo z nowymi możliwościami formularzy oraz technikami ich tworzenia. W tej wersji projektowanie formularzy to naprawdę czysta przyjemność! Ponadto zobaczysz, jak szybko i łatwo można dodać do strony elementy wideo i audio. Dodatkowo nauczysz się wykorzystywać mechanizmy do przechowywania danych oraz geolokalizacji. Dzięki tej książce poczujesz powiew świeżości w świecie stron internetowych!

- Struktura dokumentu HTML
- Struktury zasadniczych obszarów treści
- Wykorzystanie konspektu
- WAI-ARIA
- Wykorzystanie globalnych atrybutów
- Elementy do wprowadzania danych w formularzach
- Natywna obsługa multimediów
- Rozwiązywanie kłopotów z kodekami
- Obsługa wideo w starszych przeglądarkach
- Właściwości elementu Canvas
- Składowanie danych
- Metody dostępu do magazynu danych
- Praca w trybie offline
- Wykorzystanie techniki „przeciągnij i upuść”
- Geolokalizacja – możliwości oraz sposoby wykorzystania

**Twórz najnowocześniejsze witryny WWW!**

# Spis treści

Podziękowania .....	5
Wprowadzenie .....	9
<b>ROZDZIAŁ 1. GŁÓWNA STRUKTURA DOKUMENTU</b> .....	<b>17</b>
<head> .....	17
Korzystanie z nowych elementów strukturalnych HTML5 .....	22
Formatowanie HTML5 za pomocą CSS .....	25
Kiedy należy korzystać z nowych elementów strukturalnych HTML5 .....	29
Podsumowanie .....	37
<b>ROZDZIAŁ 2. TEKST</b> .....	<b>39</b>
Tworzenie struktury zasadniczych obszarów treści .....	39
Dodawanie artykułów i komentarzy na blogu .....	44
Korzystanie z konspektu .....	46
WAI-ARIA .....	63
Jeszcze więcej nowych struktur! .....	67
Elementy o zmodyfikowanej definicji .....	73
Atrybuty globalne .....	77
Własności, które nie zostały opisane w tej książce .....	80
Podsumowanie .....	81
<b>ROZDZIAŁ 3. FORMULARZE</b> .....	<b>83</b>
Kochamy HTML, a teraz jest to miłość z wzajemnością .....	83
Nowe elementy do wprowadzania danych .....	84
Nowe atrybuty .....	90
Przykład użycia nowych elementów i atrybutów .....	95
Zapewnienie zgodności wstecz ze starszymi przeglądarkami .....	98
Stosowanie stylów dla nowych pól formularzy oraz komunikaty o błędach .....	99
Przesłanie domyślnych opcji przeglądarek .....	100
Wykorzystanie JavaScriptu do zaimplementowania własnego mechanizmu walidacji .....	101
Pomijanie walidacji .....	103
Podsumowanie .....	105
<b>ROZDZIAŁ 4. WIDEO I AUDIO</b> .....	<b>107</b>
Natywna obsługa multimediów: co, jak i dlaczego? .....	107
Kodeki — prawdziwy horror .....	113
Tworzenie własnych kontroltek .....	118
Dostępność multimediów .....	126
Podsumowanie .....	129

<b>ROZDZIAŁ 5. ELEMENT CANVAS</b>	<b>131</b>
Podstawowe informacje o elemencie canvas .....	132
Rysowanie ścieżek .....	138
Transformacje: piksele w przebraniu .....	141
Przechwytywanie obrazów .....	142
Przetwarzanie pojedynczych pikseli .....	146
Animacja rysunków na elemencie canvas .....	150
Podsumowanie .....	156
<b>ROZDZIAŁ 6. SKŁADOWANIE DANYCH</b>	<b>157</b>
Opcje składowania danych .....	158
Web Storage .....	158
Web SQL Databases .....	166
Podsumowanie .....	177
<b>ROZDZIAŁ 7. TRYB OFFLINE</b>	<b>179</b>
Wyciągnięcie wtyczki: przejście do trybu offline .....	179
Manifest lokalnego bufora .....	180
W jaki sposób serwuje się manifest .....	183
Proces przeglądarka-serwer .....	184
applicationCache .....	186
Wykorzystywanie manifestu do wykrywania połączenia .....	188
Zerowanie bufora .....	189
Podsumowanie .....	190
<b>ROZDZIAŁ 8. TECHNIKA „PRZECIĄGNIJ I UPUŚĆ”</b>	<b>191</b>
Wprowadzenie w tematykę techniki „przeciągnij i upuść” .....	192
Interakcje z przeciąganymi danymi .....	195
W jaki sposób przeciągać dowolne elementy? .....	198
Dodawanie własnych ikon przeciągania .....	198
Dostępność .....	199
Podsumowanie .....	201
<b>ROZDZIAŁ 9. GEOLOKALIZACJA</b>	<b>203</b>
Lokalizowanie odwiedzających .....	203
Metody API .....	205
Jak to działa „pod maską”: prawdziwa magia .....	210
Podsumowanie .....	211
<b>ROZDZIAŁ 10. WEB MESSAGING, WEB WORKERS I WEB SOCKETS</b>	<b>213</b>
Czat z wykorzystaniem Web Messaging API .....	214
Wielowątkowość z wykorzystaniem interfejsu Web Workers .....	216
Web Sockets: praca ze strumieniami danych .....	226
Podsumowanie .....	230
I na koniec... .....	231
<b>SKOROWIDZ</b>	<b>233</b>

# Geolokalizacja

Remy Sharp

**API GEOLOKALIZACJI** to jeden z tych API, które nie mają zupełnie nic wspólnego ze specyfikacją HTML5. Został opracowany przez organizację W3C, a nie WHATWG. W rzeczywistości ten interfejs API nigdy nie był częścią oryginalnej specyfikacji Web Application (choć obecnie odwołuje się do specyfikacji HTML5). Jest jednak tak interesujący, że musieliśmy umieścić jego opis w niniejszej książce. W gruncie rzeczy jest to kluczowy API dla aplikacji, które mają zostać wyposażone w super „odlotowe” efekty i mechanizmy interakcji.

API jest niewiarygodnie prosty. Za jego pomocą można z łatwością usprawnić dowolne aplikacje wykorzystujące dane geograficzne. Wystarczy włączyć to API do aplikacji — to wszystko, czego potrzeba, aby zwolnić użytkowników z obowiązku poruszania się „palcem po mapie”.

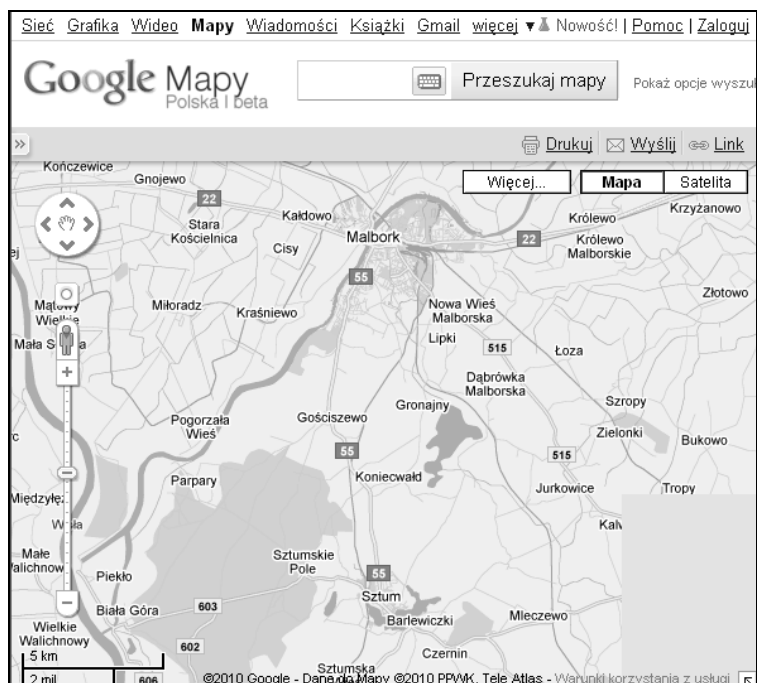
Przeglądarka Firefox 3.5+ to obecnie jedyna przeglądarka desktop (w wersji produkcyjnej) obsługująca API geolokalizacji. Przeglądarka Chrome zawiera jednak obsługę tego API w kompilacji deweloperskiej, zatem można się spodziewać, że wkrótce zostanie wyposażona w ten interfejs. Chociaż API nie jest powszechnie wspierany w przeglądarkach desktop, wspiera go wiele przeglądarek działających w telefonach komórkowych. W szczególności chodzi o implementację WebKit dla urządzeń przenośnych. O dziwo wersja desktop pakietu WebKit nie obsługuje API geolokalizacji. Jestem jednak przekonany, że ma to związek z coraz częstszym wsparciem dla technologii GPS w telefonach komórkowych. Fennec, przeglądarka na telefony komórkowe firmy Mozilla, również wspiera geolokalizację. W interfejs API geolokalizacji jest również wyposażony PhoneGap — framework do tworzenia aplikacji Open Web — zatem jest on dostępny także dla tych programistów, którzy tworzą swoje aplikacje w oparciu o technologie Open Web.

## Lokalizowanie odwiedzających

API geolokalizacji daje nam sposób wyznaczania dokładnej pozycji użytkownika odwiedzającego nasz serwis. W internecie dostępnych jest już bardzo wiele aplikacji wykorzystujących ten API — począwszy od obsługi map w systemie Google Maps

(rysunek 9.1), a skończywszy na bazujących na lokalizacji serwisach społecznościowych, takich jak Gowalla czy Google Buzz.

**RYSUNEK 9.1.** Google Maps wykrywa obsługę geolokalizacji i dodaje funkcjonalność „zlokalizuj mnie”



API geolokalizacji oferuje dwie metody pozwalające na uzyskiwanie informacji geograficznych od użytkownika:

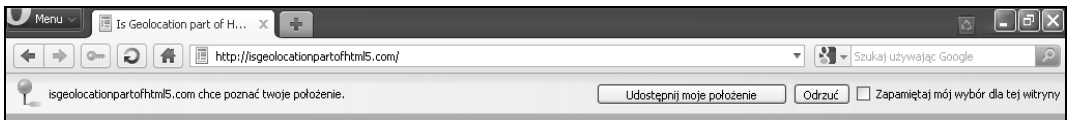
1. `getCurrentPosition` to metoda, która pozwala „jednorazowo” pobrać bieżącą lokalizację użytkownika.
2. Metoda `watchPosition` śledzi pozycję użytkownika. W celu sprawdzenia, czy nie uległa zmianie, dopytuje o nią w regularnych odstępach czasu.

Zarówno metoda `getCurrentPosition`, jak i `watchPosition` ustalają lokalizację użytkownika w sposób asynchroniczny. Jest jeden wyjątek do tej reguły: kiedy użytkownik odwiedza serwis po raz pierwszy, przeglądarka wyświetla modalne okno dialogowe z pytaniem o zezwolenie na śledzenie jego lokalizacji. W przypadku gdy aplikacja wykorzystuje dowolną z wymienionych metod, większość urządzeń wyświetla pytanie o zgodę na śledzenie ich lokalizacji. Jeśli nie wyrażą zgody na udzielanie informacji o swojej lokalizacji, to można tak skonfigurować API, aby informacje o błędach były przekazywane do wskazanej funkcji.

Oto, co mówi specyfikacja:

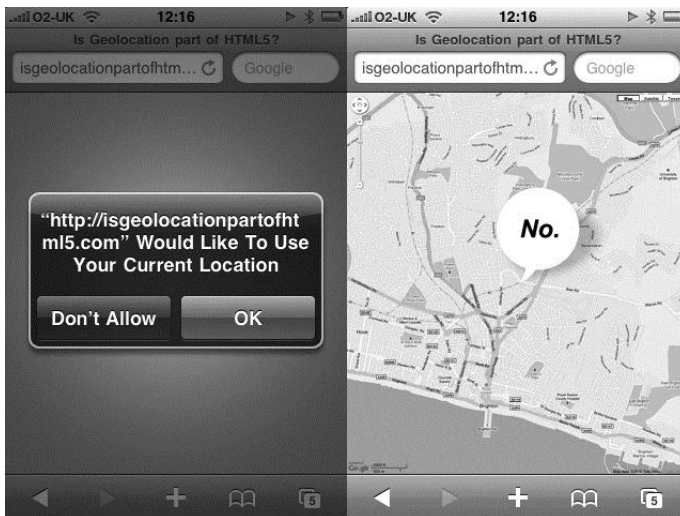
„Agenty użytkownika nie mogą wysyłać informacji o lokalizacji do serwisów bez zgody udzielonej przez użytkownika”.

A zatem przeglądarka ma obowiązek poinformować użytkowników, że serwis próbuje uzyskać informacje o ich bieżącej pozycji. Różne przeglądarki obsługują to w różny sposób. Przeglądarka Firefox oferuje na przykład niemodalne, nieblokujące ostrzeżenie z pytaniem do użytkowników o zgodę na współdzielenie lokalizacji (patrz rysunek 9.2). Oznacza to, że w czasie gdy wyświetla się to ostrzeżenie, aplikacja kontynuuje działanie.



**RYСУNEK 9.2.** Przeglądarka Firefox wyświetla pytanie o zgodę użytkownika na śledzenie jego lokalizacji

Inne przeglądarki, na przykład Mobile Safari, wyświetlają modalne okno dialogowe. Wykonywanie kodu aplikacji jest zatrzymywane do chwili, kiedy użytkownik odpowie (rysunek 9.3).



**RYСУNEK 9.3.** Przeglądarka Mobile Safari wyświetla modalne okno dialogowe, zatrzymując wykonywanie aplikacji

## Metody API

API geolokalizacji istnieje wewnątrz obiektu `navigator` i zawiera tylko trzy metody:

- ❖ `getCurrentPosition`,
- ❖ `watchPosition`,
- ❖ `clearWatch`.

Metody `watchPosition` i `clearWatch` tworzą parę. Metody `watch` i `clear` działają w taki sam sposób, jak metody `setInterval` i `setTimeout`. Metoda `watchPosition` zwraca unikatowy identyfikator, który jest przekazywany do metody `clearWatch` w celu usunięcia określonej obserwacji.

Jak wspomniałem wcześniej, metody `getCurrentPosition` i `watchPosition` działają podobnie, dlatego pobierają takie same argumenty:

- ❖ procedurę obsługi sukcesu,
- ❖ procedurę obsługi błędów,
- ❖ opcje geolokalizacji.

Najprostsze użycie API geolokalizacji może polegać na wysłaniu nazwy procedury obsługi sukcesu do metody `getCurrentPosition`:

```
navigator.geolocation.getCurrentPosition(function (position) {
    alert('Znaleźliśmy Cię!');
    // wykonaj jakieś działania z danymi o pozycji
});
```

## Mamy Cię: procedura obsługi sukcesu

Jeśli użytkownik zezwoli przeglądarce na udzielanie informacji o jego lokalizacji i nie wystąpi jakiś inny błąd, to przeglądarka uruchomi procedurę obsługi sukcesu. Jej nazwa jest pierwszym argumentem metod `getCurrentPosition` i `watchPosition`.

Do tej procedury jest przekazywany obiekt `Position` posiadający dwie właściwości: `coords` (obiekt zawierający informacje o współrzędnych) oraz `timestamp`. Najbardziej interesujące informacje są zapisane w obiekcie `coords`. Obiekt `Position` zawiera dwie klasy informacji. Dane pierwszej klasy występują we wszystkich przeglądarkach z obsługą geolokalizacji: Chrome 5, Firefoksie 3.5+, Mobile Safari i Safari 5 oraz w Androidzie 2.0. Są to:

- ❖ `readonly attribute double latitude`,
- ❖ `readonly attribute double longitude`,
- ❖ `readonly attribute double accuracy`.

Warto zwrócić uwagę, że atrybut `accuracy` to miara dokładności pomiaru długości i szerokości geograficznej wyrażona w metrach. Informację tę można wykorzystać do pokazania promienia dokładności w przypadku prezentowania pozycji użytkownika na mapie.

Istnieje także druga klasa danych wewnątrz obiektu `coords`, ale obecnie do tych atrybutów nie są przypisane żadne wartości. We wszystkich współczesnych przeglądarkach zawierających natywną obsługę geolokalizacji atrybuty te będą miały wartość `null`, `0` lub `NaN`. Są to następujące atrybuty:



- ❖ `readonly attribute double altitude,`
- ❖ `readonly attribute double altitudeAccuracy,`
- ❖ `readonly attribute double heading,`
- ❖ `readonly attribute double speed.`

Wykorzystując dane o współrzędnych, można z łatwością odwzorować bieżącą pozycję użytkownika na mapie:

```
if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(function
        (position) {
            var coords = position.coords;
            showMap(coords.latitude, coords.longitude,
                coords.accuracy);
        });
}
```

W wielu aplikacjach użytkownik ma do dyspozycji ręczny sposób ustawiania swojej bieżącej pozycji. Jeśli jest dostępna metoda bazująca na geolokalizacji, to witryna może oferować zaawansowaną funkcjonalność polegającą na progresywnym ulepszeniu strony lub ogólnej prezentacji.

Przykład takiej aplikacji można obejrzeć pod adresem <http://owlsnearyou.com>. Jeśli przeglądarka wspiera API geolokalizacji, to po odwiedzeniu serwisu następuje odświeżenie strony i wyświetlenie miejsca, w którym użytkownik może polować na sowy (patrz rysunek 9.4), jeśli to właśnie jest jego zajęcie po całym dniu tworzenia aplikacji HTML5. Jeśli geolokalizacja nie jest dostępna, serwis wyświetla prośbę o podanie lokalizacji.

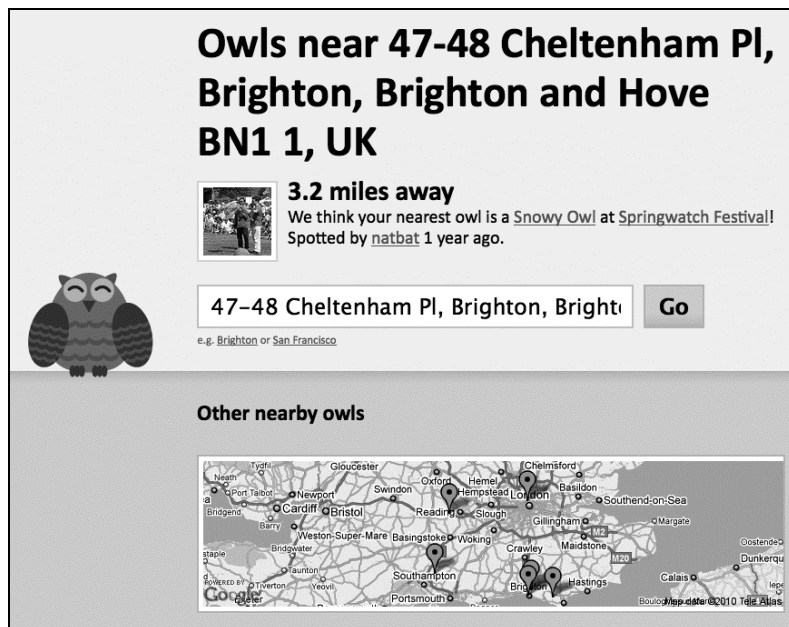
## Geo 404: procedura obsługi błędów

Drugim argumentem metod `getCurrentPosition` i `watchPosition` jest nazwa procedury obsługi błędów. Jest to szczególnie ważne w sytuacji, kiedy chcemy dostarczyć alternatywny sposób wskazywania lokalizacji (na przykład ręcznie) albo powiadomić użytkownika o wystąpieniu błędów podczas ustalania jego pozycji. Procedura obsługi błędów może zostać wywołana, w przypadku gdy użytkownik nie zgodzi się na podanie pozycji. Może się jednak zdarzyć, że użytkownik zezwolił na śledzenie jego pozycji i aplikacja działająca w telefonie komórkowym ją śledzi, ale telefon chwilowo znalazł się poza zasięgiem sieci. Taka sytuacja także spowoduje wywołanie procedury obsługi błędów.

Procedura obsługi błędów otrzymuje jeden argument z obiektem opisującym błąd pozycji. Obiekt ten ma dwie właściwości:

- ❖ `readonly attribute unsigned short code,`
- ❖ `readonly attribute DOMString message.`

**RYSUNEK 9.4.** Przykład progresywnego ulepszania interfejsu z wykorzystaniem geolokalizacji



**Owls near 47-48 Cheltenham Pl, Brighton, Brighton and Hove BN1 1, UK**

**3.2 miles away**  
We think your nearest owl is a [Snowy Owl](#) at [Springwatch Festival!](#)  
Spotted by [natbat](#) 1 year ago.

47-48 Cheltenham Pl, Brighton, Bright

e.g. Brighton or San Francisco

**Other nearby owls**

Właściwość `code` może przybrać jedną z następujących wartości:

- ❖ `PERMISSION_DENIED` (wartość liczbowa 1),
- ❖ `POSITION_UNAVAILABLE` (wartość liczbowa 2),
- ❖ `TIMEOUT` (wartość liczbowa 3).

Właściwość `message` jest przydatna podczas debugowania, ale nie powinno się ujawniać jej użytkownikowi. Właściwość `message` nie zawsze jest dostępna (np. nie jest obecnie dostępna w przeglądarce Firefox 3.5+).

Dla przykładu gdybyśmy użyli poniższego kodu, a użytkownik nie zgodziłby się na informowanie o jego pozycji, wyświetliłaby się strona z komunikatem „Brak uprawnień: nie możemy określić Twojej pozycji”.

```
if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(function
    (position) {
    var coords = position.coords;
    showMap(coords.latitude, coords.longitude,
      coords.accuracy);
  }, function (error) {
    var errorTypes = {
      1: 'Brak uprawnień',
      2: 'Pozycja niedostępna',
      3: 'Upłynął limit czasu żądania'
```

```

    };

    alert(errorTypes[error.code] + ": nie możemy
    określić Twojej pozycji");
  });
}

```

### INNE BŁĘDY: PO POMYŚLNYM USTALENIU LOKALIZACJI

Innym przypadkiem, gdy należałoby wywołać procedurę obsługi błędów, jest sytuacja, w której w procedurze obsługi sukcesu okazuje się, że wartość atrybutu **accuracy** jest ustawiona na bardzo dużą wartość. Kiedyś, pracując w domu w Brighton na południu Anglii, odwiedzałem stronę, która miała określić moją lokalizację. O dziwo umiejscowiła mnie w centrum Londynu! Kiedy sprawdziłem log za pomocą konsoli Firebug, zobaczyłem, że dokładność żądania geolokalizacji ustawiono na 140 000 metrów — trzeba przyznać, iż 140 kilometrów to dość duży promień niedokładności! Nie dziwię się, że aplikacja nie miała pewności, gdzie dokładnie jestem. Z tego powodu gorąco polecam, aby podczas tworzenia aplikacji wykorzystującej geolokalizację sprawdzać także atrybut **accuracy** w procedurze obsługi sukcesu. Jeśli atrybut ten ma nadzwyczajnie dużą wartość, lepiej zignorować uzyskane informacje o pozycji i pozwolić użytkownikowi podać swoją pozycję ręcznie.

## Konfiguracja geolokalizacji

Trzecim i ostatnim argumentem metod `getCurrentPosition` i `watchPosition` są opcje geolokalizacji. Wszystkie parametry są *nieobowiązkowe* i zawierają następujące informacje:

- ❖ `enableHighAccuracy` (typu `Boolean`, domyślnie `false`),
- ❖ `timeout` (w milisekundach, domyślnie nieskończoność — liczba 0),
- ❖ `maximumAge` (w milisekundach, domyślnie 0).

Na przykład, aby zażądać określania pozycji z dużą dokładnością, dwusekundowym limitem czasu oraz by nigdy nie używać przestarzałych danych geograficznych, możemy wywołać metodę `getCurrentPosition`, używając poniższych opcji (`success` i `error` są predefiniowanymi funkcjami):

```

navigator.geolocation.getCurrentPosition(success, error, {
    enableHighAccuracy: true,
    timeout: 2000,
    maximumAge: 0
});

```

Opcja `enableHighAccuracy` nakazuje urządzeniu podjęcie próby uzyskania dokładniejszego odczytu długości i szerokości geograficznej. W przypadku urządzeń przenośnych może to oznaczać wykorzystanie modułu GPS, co wiąże się z większym zużyciem energii. Zwykle chcemy oszczędzać baterie tak długo, jak się da, dlatego domyślnie atrybut `enableHighAccuracy` jest ustawiony na `false`.

Atrybut `timeout` informuje procedurę geolokalizacyjną o tym, jak długo powinien czekać na pozycję, zanim zrezygnuje i wywoła procedurę obsługi błędu (odliczanie nie rozpoczyna się jednak, jeśli aplikacja czeka na potwierdzenie żądania przez użytkownika). W przypadku upływu limitu czasu kod błędu jest ustawiany na wartość 3 (`TIMEOUT`). Ustawienie parametru `timeout` na 0 (obecnie wartość domyślna) oznacza oczekiwanie w nieskończoność.

Ostatnią dostępną opcją jest `maximumAge`, którą można wykorzystać do poinformowania przeglądarki o tym, czy mają być używane dane o pozycji, które zostały wcześniej zbuforowane. Jeśli żądanie nastąpi w czasie odległym o nie więcej niż `maximumAge` (w milisekundach), to zostanie zwrócona informacja z bufora i nie będzie wysłane nowe żądanie o ustalenie pozycji. Wartością opcji `maximumAge` może być również `Infinity`, co jest poleceniem dla przeglądarki, by *zawsze* zwracała pozycję z bufora. Ustawienie parametru `maximumAge` na 0 (wartość domyślna) oznacza, że przeglądarka musi ustalać nową pozycję przy każdym żądaniu.

## Jak to działa „pod maską”: prawdziwa magia

API geolokalizacji wykorzystuje kilka różnych technik ustalania pozycji. Większość z tych technik to dla zwykłych ludzi (włącznie ze mną) prawdziwa czarna magia. Warto jednak mieć pogląd na temat tego, co znajduje się „pod maską”, ponieważ ma to wpływ na dokładność uzyskiwanych danych o pozycji.

Jednym z oczywistych sposobów uzyskiwania danych o pozycji jest system GPS. W funkcjonalność GPS jest wyposażonych coraz więcej urządzeń — począwszy od telefonów komórkowych, a skończywszy na laptopach. Przy założeniu, że mamy dobry kontakt ze stacją naziemną GPS (która zbiera odczyty z satelitów w celu triangulacji naszej pozycji — coraz więcej czarnej magii), uzyskujemy bardzo dokładne informacje na temat położenia. GPS powinien również podać informacje o wysokości (jak widzieliśmy, wysokość należy do drugiej klasy właściwości obiektu opisu współrzędnych). Na razie takie dane nie wyświetlają się w przeglądarkach. Spodziewam się jednak, że z czasem staną się one dostępne, ponieważ technika urządzeń mobilnych szybko się rozwija.

Inną metodą ustalania pozycji jest wykorzystanie informacji z sieci. Jest to typowy sposób stosowany w przeglądarkach desktop, na przykład w Firefoksie. Technika ta obejmuje wykorzystanie triangulacji WIFI i adresów IP do próby jak najdokładniejszego określenia lokalizacji. Programista wywołuje metodę API geolokalizacji z przeglądarki, a przeglądarka odwołuje się do serwisu zewnętrznego, na przykład Skyhook lub Google. Chociaż sposób ten nie gwarantuje takiej dokładności jak GPS (licząc

w metrach), to może to być bardzo dobre rozwiązanie zastępcze. Trzeba bowiem pamiętać, że system GPS nie działa dobrze w pomieszczeniach lub obszarach silnie zurbanizowanych.

Ogólnie rzecz biorąc, nie jest szczególnie istotne, by wiedzieć, co sprawia, że geolokalizacja działa. Jeśli jednak chcemy uzyskać większą dokładność, powinniśmy skorzystać z bardziej energożernych urządzeń (takich jak GPS) i godzić się na to, że baterie w urządzeniach szybciej się wyczerpią.

Podsumowując, geolokalizacja to bardzo fajna czarna magia.

## Podsumowanie

W Wielkiej Brytanii, gdzie ciągle jeszcze często spotyka się czerwone budki telefoniczne, w każdej z nich można było znaleźć tabliczkę z nadrukowanym adresem budki. Dzięki temu, dzwoniąc na policję lub pogotowie, można było podać swoją pozycję. Oczywiście pomagało to również zagubionym osobom po nocnej imprezie. Dzwoniąc rano do domu z prośbą o transport, można było powiedzieć, gdzie się jest.

Przeglądarki mają teraz wbudowaną tę funkcję. Jest to jeden z najprostszych API do zaimplementowania. Jeśli serwis WWW potrzebuje od użytkowników dowolnych informacji na temat lokalizacji, to z łatwością można usprawnić swoją stronę, wyposażając ją w wywołania nowego API. Jest tylko kwestią czasu, kiedy powstaną takie serwisy WWW, które po otwarciu natychmiast poinformują nas, gdzie jest najbliższa ulubiona kawiarnia lub restauracja, bazując na bieżącej porze dnia oraz miejscu, w którym się znajdziemy.

# Skorowidz

.stringify, 163  
:heading(), 51  
:invalid, 99  
:-moz-any(), 51  
:required, 99  
<!doctype html>, 17  
<address>, 74  
<applet>, 76  
<article>, 24, 26, 36, 40, 44,  
52, 53, 54, 55, 62  
<section>, 56  
<aside>, 34, 46, 68  
<audio>, 108, 114  
monitorowanie postępu  
pobierania, 123  
zdarzenia, 122  
<b>, 75  
<big>, 76  
<blink>, 76  
<blockquote>, 50  
stopka, 44  
<body>, 50  
<canvas>, 109, 132, 133  
<center>, 76  
<cite>, 74  
<command>, 81  
<content>, 26  
<datalist>, 90  
<details>, 50, 68  
<device>, 109  
<div>, 23, 24, 40, 46  
draggable, 198  
<dl>, 73  
<em>, 69, 75  
<embed>, 76, 80, 108  
<fieldset>, 50  
<figcaption>, 69  
<figure>, 50, 69  
<font>, 76  
<footer>, 25, 26, 35, 41, 43  
<form>, 95  
<head>, 17  
<header>, 24, 26, 29, 43  
<hgroup>, 49, 50  
<hr>, 76  
<html>, 19, 20  
lang, 21  
<hx>, 52  
<i>, 75  
<iframe>, 76  
<input>, 84  
list, 90  
max, 94  
min, 94  
multiple, 92  
pattern, 92, 93  
step, 95  
type=color, 89  
type=date, 86  
type=datetime, 87  
type=email, 85  
type=month, 87  
type=number, 88  
type=range, 88, 97  
type=search, 89  
type=tel, 89  
type=text, 84  
type=time, 87  
type=url, 85  
type=week, 87  
<keygen>, 80  
<mark>, 69, 71  
<marquee>, 76  
<media>, 118  
<menu>, 81  
<meta>, 18  
<meter>, 80, 96  
<nav>, 24, 26, 31, 32  
grupowanie  
elementów, 33  
<object>, 108, 109  
<ol>, 73  
start, 73  
<output>, 96  
<progress>, 80, 96  
<rp>, 71, 72  
<rt>, 71, 72  
<ruby>, 71, 72  
<section>, 46, 47, 52, 54  
<article>, 56  
<small>, 35, 40, 76  
<source>, 114  
<strong>, 75  
<style scoped>, 81  
<td>, 50  
<textarea>, 84  
<time>, 41, 42  
datetime, 42  
<track>, 126  
<video>, 107, 108  
autoplay, 111  
controls, 111  
height, 112  
kontrolki sterujące, 111  
loop, 112  
monitorowanie postępu  
pobierania, 123  
poster, 112  
preload, 112  
src, 110, 113  
style CSS, 109  
width, 112  
zdarzenia, 122

## A

AAC, 115  
addEventListener(), 122,  
123, 124  
Adobe Flash, 14  
adres e-mail, 85  
adres URL, 85  
Ajax, 13  
akcent, 75  
akcje formularzy, 83  
algorytm Scooby Doo, 26  
algorytm tworzenia  
konspektu, 46, 47  
allowedEffect, 201  
animacja rysunków na  
elementach canvas, 150  
API elementu canvas, 131  
API trybu  
natychmiastowego, 140  
API trybu opóźnionego, 140  
aplikacje internetowe, 14  
aplikacje offline, 180  
aplikacje Open Web, 203  
Apple VoiceOver, 67  
application, 65  
applicationCache, 180, 186,  
187  
arc, 136, 138, 139  
ARIA, 64  
aria-\*, 78  
aria-grabbed, 201  
aria-hidden, 78  
aria-required, 92  
aria-valuemax, 97  
aria-valuemin, 97  
aria-valuenow, 97, 98  
arkusze CSS, 23  
article, 36, 44, 52, 65  
artykuły, 40  
atrybuty  
aria-\*, 64, 78  
aria-valuemax, 97  
aria-valuemin, 97  
aria-valuenow, 97, 98  
atrybuty danych  
użytkownika, 128  
atrybuty globalne, 77  
autocomplete, 94  
autofocus, 91  
class, 22  
contenteditable, 77  
contextmenu, 77  
data-\*, 78, 128  
draggable, 78, 198  
form, 83  
formnovalidate, 103

- atrybuty
    - hidden, 78
    - id, 83
    - item, 78
    - itemprop, 78
    - list, 90
    - manifest, 180
    - max, 94
    - min, 94
    - multiple, 92
    - name, 95
    - novalidate, 103
    - pattern, 92
    - placeholder, 91
    - playbackRate, 125
    - preload, 125
    - pubdate, 42
    - required, 92, 100
    - role, 78, 97
    - spellcheck, 79
    - step, 94
    - subject, 78
    - tabindex, 79
    - validity, 102
  - Audacity, 117
  - audio, 107
  - autocomplete, 94
  - autofocus, 91
  - automatyczne odtwarzanie
    - klipu wideo
    - lub audio, 111
  - autoplay, 111
- B**
- banner, 65
  - baza danych, 166
    - odpytywanie tabeli, 170
    - otwieranie, 168
    - transakcje, 175
    - tworzenie, 168
    - wstawianie danych, 170
  - beginPath, 138, 139
  - block-level, 70
  - blog, 39
    - artykuły, 44
    - komentarze, 44
    - wpisy, 44
    - wprowadzanie
      - komentarzy, 95
  - breadcrumb trails, 32
- C**
- CACHE, 183
  - CACHE MANIFEST, 181
  - cachebusting, 183
  - Camino, 28
  - canPlayType(), 118
  - canvas, 131, 132, 219
    - animacja rysunków, 150
    - API, 133, 134
    - clearRect, 142
    - czyszczenie
      - elementu, 142
    - dostępność, 155
    - drawImage, 143
    - eksport bieżącego
      - stanu, 149
    - fillStyle, 135, 137, 155
    - fillText, 155
    - filtry, 146
    - getImageData, 146
    - gradient, 135
    - gry, 155
    - moveTo, 139
    - oderwanie ołówka od
      - papieru, 139
    - odtwarzanie stanu
      - rysowania, 153
    - origin-clean, 148
    - processing.js, 150
    - przechwytywanie
      - obrazów, 142
    - przechwytywanie
      - obrazu z elementu
        - video, 143
    - przetwarzanie pikseli, 146
    - regiony obcinania, 153
    - renderowanie tekstu, 154
    - restore, 153
    - rotate, 141
    - save, 153
    - stan rysowania, 153
    - styl wypełnienia, 135
    - ścieżki, 138
    - toDataURL, 149, 150
    - transformacje, 141, 153
    - translate, 141
    - URL data, 149
    - wparcie w
      - przeglądarkach, 134
    - współrzędne, 138
    - wzorce, 135
    - zapisywanie do pliku, 149
    - zapisywanie stanu
      - rysowania, 153
    - zastępowanie tła
      - wideo, 143
  - CanvasPixelArray, 146
  - cele tworzenia języka
    - HTML5, 13
  - charset, 18
  - checkValidity, 102
  - ciasteczka, 157
  - class, 22, 23
  - clear, 162
  - clearInterval, 144
  - clearRect, 142, 153
  - clearWatch, 205
  - clipping regions, 153
  - close, 218
  - color, 89
  - Comet, 213, 226
  - complementary, 65
  - contenteditable, 77
  - contentEditable, 77
  - contentinfo, 65
  - contentWindow, 214
  - contextmenu, 77
  - control.focus(), 91
  - controls, 111
  - coords, 206
  - CREATE TABLE, 170, 171
  - createPattern, 135, 136, 137
  - CSS, 23, 25
  - cudzysłowy, 18
  - currentTime, 143
  - customError, 102
  - cytaty, 68
  - czas, 41, 42, 87
  - czat, 214, 229
  - czynniki ekranu, 67
- D**
- dane o prawach
    - autorskich, 35
  - data-\*, 78, 128
  - data-begin, 128
  - data-finish, 128
  - dataset, 128
  - data-shields, 128
  - data-start, 128
  - dataTransfer, 195
  - date, 86
  - datetime, 42, 87
  - daty, 41, 86
  - degradacja z wdziękiem, 91
  - delete, 83
  - diagram strony
    - wersja HTML4, 23
    - wersja HTML5, 25
  - Digital Right
    - Management, 109
  - display:block, 26
  - display:inline, 26, 70
  - DOCTYPE, 17
  - document, 65
  - document.cookie, 157
  - document.createElement(), 28, 98, 137
  - document.designMode, 77
  - dokumenty ARIA, 65
  - dokumenty HTML5, 17
  - DOM, 14, 21
  - DOM inspector, 98

dostępność, 52  
 multimedia, 126  
 technika przeciągnij  
 i upuść, 199  
 dragend, 201  
 draggable, 78, 198  
 DragonFly, 164  
 DragOver, 192  
 dragstart, 195, 198  
 drawImage, 143, 145  
 drawing context, 131  
 DRM, 109  
 drobny maczek, 35  
 dropEffect, 201  
 dropover, 194  
 dwukierunkowe  
 połączenie między  
 serwerem a klientem, 226

## E

efekt zajawek, 58  
 efekty typograficzne, 68  
 eksperymentalny parser  
 HTML5, 15  
 element.validationMessage,  
 102  
 elementy do wprowadzania  
 danych, 84  
 elementy interaktywne, 70  
 elementy nagłówkowe, 70  
 elementy osadzone, 70  
 elementy poziomu bloku, 70  
 elementy poziomu wiersza,  
 70  
 elementy strukturalne, 22,  
 29  
 email, 85  
 end(), 123  
 EST, 42  
 event, 215  
 event.preventDefault(),  
 193, 194  
 executeSql, 169, 170, 173  
 expandos, 161

## F

FALLBACK, 182, 183, 188  
 ff mpeg, 117  
 fieldset, 104  
 getElementsByTagName,  
 104  
 fill, 138, 139  
 fillRect, 133, 134  
 fillStyle, 134, 135, 136, 137,  
 139, 155  
 fillText, 155  
 Firebug, 164

Firefogg, 117  
 Firefox, 28, 54  
 tryb offline, 185  
 Flash Player, 116  
 flow, 70  
 focus(), 79  
 footer, 35  
 forEach, 171  
 form, 65  
 format czasu, 42  
 formatowanie strony, 23, 25  
 formnovalidate, 103  
 formularze, 83  
 akcje, 83  
 atrybuty, 90  
 elementy, 83  
 elementy  
 wprowadzania  
 danych, 84  
 implementacja  
 mechanizmu  
 walidacji, 101  
 kalendarz, 86  
 komunikaty o błędach, 99  
 oninputchange, 104  
 pomijanie walidacji, 103  
 powiązanie  
 z formularzem, 83  
 przesłanie  
 domyślnych opcji  
 przeglądarek, 100  
 stosowanie stylów dla  
 nowych pól, 99  
 typy formularzy, 84  
 wprowadzanie  
 komentarzy na  
 blogu, 95  
 wymuszanie walidacji  
 elementów, 102  
 wyświetlanie rezultatów  
 obliczeń, 96  
 zgodność ze starszymi  
 przeglądarkami, 98  
 fraza, 70

## G

geolokalizacja, 203  
 accuracy, 206  
 API, 204  
 błędy, 209  
 clearWatch, 205  
 coords, 206  
 dane o współrzędnych,  
 207  
 dokładność pomiaru,  
 206  
 enableHighAccuracy,  
 209

getCurrentPosition, 204,  
 205, 206, 209  
 GPS, 210  
 konfiguracja  
 geolokalizacji, 209  
 lokalizowanie  
 odwiedzających, 203  
 navigator, 205  
 określanie pozycji z  
 dużą dokładnością,  
 209  
 PERMISSION\_DENIED,  
 208  
 Position, 206  
 POSITION\_UNAVAILABL  
 E, 208  
 procedura obsługi  
 błędów, 207  
 procedura obsługi  
 sukcesu, 206  
 techniki ustalania  
 pozycji, 210  
 timeout, 210  
 TIMEOUT, 208  
 timestamp, 206  
 watchPosition, 204, 205,  
 206, 209

get, 83  
 getAttribute, 128  
 getContext, 133  
 getCurrentPosition, 204,  
 205, 206  
 getData, 195  
 getElementById, 172  
 getImageData, 146  
 GIF, 149  
 główne elementy podziału  
 na sekcje, 50  
 gniazda, 213, 227  
 Google Buzz, 204  
 Google Maps, 204  
 Google Suggest, 90  
 Google Wave, 227  
 Gowalla, 204  
 GPS, 210  
 graceful degradation, 91  
 gradient, 135, 136  
 grafika 2D, 131  
 green screen replacement,  
 143  
 grupowanie elementów  
 <nav>, 33

## H

H.264, 114, 115  
 header, 29  
 heading, 65  
 height, 112



hidden, 78  
 historia HTML5, 10  
 href, 54  
 HTML, 9  
 HTML5, 9, 10  
   elementy usunięte, 76  
 HTML5Element, 128  
 HTMLFieldSetElement.elements, 104  
 http-equiv, 18

## I

id, 23, 83  
 IE Print Protector, 28  
 iframe, 214  
 ikony przeciągania, 198  
 immediate mode API, 140  
 importScripts, 222  
 informacje kontaktowe  
   autora, 74  
 INSERT, 170, 171  
 Internet Explorer, 27  
   style CSS, 27  
 Internet Explorer 8, 28  
 Internet Explorer 9, 28  
 INVALID\_STATE\_ERROR, 168  
 isContentEditable, 77  
 item, 78  
 itemprop, 78

## J

JavaScript, 14  
   mechanizm walidacji,  
     101  
 JAWS, 52  
 język dokumentu, 20  
 język HTML5, 9, 10  
 języki azjatyckie, 71  
 JIT, 155  
 JPG, 149  
 jQuery Visualize, 155  
 JSON, 163, 197, 215  
 JSON.parse, 163, 216  
 JSON.stringify, 215, 216

## K

kalendarz, 86  
 kategorie elementów, 70  
 key, 162  
 klip wideo, 108  
 kodeki, 113  
   H.264, 114, 115  
   MP3, 114  
   Ogg Theora, 113  
   Ogg Vorbis, 113

Theora, 114  
 VP8, 114  
 wiele elementów  
   <source>, 114  
 kodowanie JSON, 197  
 kodowanie znaków, 18  
 kolory, 89  
 kompresja wideo, 116  
 komunikaty, 213  
   wysyłanie, 214  
 komunikaty o błędach, 99  
 konspekt, 46  
   <hgroup>, 49  
   algorytm tworzenia, 47  
   dostępność, 52  
   główne elementy  
     podziału na sekcje, 50  
   sekcje <aside>, 48  
   sekcje <nav>, 48  
   sekcje bez tytułu, 48  
   style CSS, 51  
 kontekst rysowania, 131  
 kontrolki sterowania  
   multimediami, 118  
 konwersja stopni  
   na radiany, 136

## L

lang, 20  
 lazy loading, 98  
 length, 123  
 leniwe ładowanie, 98  
 lineWidth, 134, 139  
 linki „prawne”, 32  
 list, 90  
 listy, 33  
 listy asocjacji, 73  
 load, 137  
 loadeddata, 124, 144  
 loadedmetadata, 125  
 localStorage, 159, 164  
 lokalizowanie  
   odwiedzających, 203  
 loop, 112

## Ł

łącza poziomego bloku, 54  
 łączenie ARIA z HTML5, 66  
 łuk, 139

## M

main, 65  
 manifest, 179, 180  
 manifest lokalnego  
   bufora, 180

max, 94  
 mechanizm nawigacji, 31  
 mechanizm walidacji, 101  
 media, 118, 119  
 menu kontekstowe, 81  
 metadata, 70  
 microdata, 81  
 Microsoft Silverlight, 14  
 MIDI, 107  
 miernik postępu, 80  
 MIME XML, 15  
 min, 94  
 Miro Video Converter, 117  
 mod\_expires, 185  
 model DOM, 14, 21  
 model treści HTML5, 70  
 monitorowanie postępu  
   pobierania, 123  
 month, 87  
 moveTo, 139, 141  
 MP3, 107, 114  
 multimedia, 107  
 multiple, 85, 92

## N

nagłówek, 29, 43  
 name, 95  
 natywna obsługa  
   multimediałów, 107  
 navigation, 65  
 navigator, 205  
 navigator.onLine, 188  
 nawiasy ruby, 72  
 nawigacja, 31  
   okruszki chleba, 32  
 NETWORK, 183  
 notacja JSON, 163  
 novalidate, 103  
 number, 88  
 numer telefonu, 89  
 numer tygodnia, 87  
 NVDA, 67

## O

obsługa HTML5, 15  
 obsługa multimediałów, 107,  
   113  
 obsługa wideo w starszych  
   przeglądarkach, 115  
 obszar zasadniczej treści,  
   61, 66  
 obszary treści, 39  
 odruch wymiotny, 54  
 odtwarzanie filmu, 120  
 Ogg Theora, 113, 117  
 Ogg Vorbis, 113  
 OggConvert, 117

okruszki chleba, 32  
 ondragover, 193  
 ondrop, 193  
 OnDrop, 192  
 onforminput, 96, 105  
 oninputchange, 104  
 onmessage, 218  
 opcje składowania danych,  
 158  
 Open Web, 203  
 openDatabase, 168  
 Opera Dragonfly, 19  
 origin-clean, 148  
 otwieranie baz danych, 168

## P

panel boczny, 33, 34, 68  
 paski narzędzi, 81  
 pattern, 92  
 patternMismatch, 102  
 pause(), 118, 122  
 pauza filmu, 120  
 phrasing, 70  
 piksele, 141  
 placeholder, 91  
 play(), 118, 122  
 playbackRate, 125  
 PNG, 149  
 podpis ilustracji, 69  
 podtytuł, 49  
 podział na sekcje, 70  
 polyfilling, 91  
 połączenie pomiędzy  
 serwerem a klientem,  
 226  
 pomijanie walidacji, 103  
 Position, 206  
 post, 83  
 poster, 112  
 postMessage, 214, 215, 217  
 powiązanie podpisu z  
 ilustracją, 69  
 powiązanie z formularzem,  
 83  
 prawa autorskie, 35  
 preload, 112, 113, 125  
 auto, 113  
 metadata, 113  
 none, 113  
 procedury nasłuchiwania  
 zdarzeń, 104  
 processing.js, 150  
 przechwytywanie obrazów,  
 142  
 przeciągnij i upuść, 191  
 allowedEffect, 201  
 ARIA, 200  
 aria-grabbed, 201

czytniki ekranów, 201  
 dataTransfer, 195  
 dostępność, 199  
 draggable, 198  
 DragOver, 192  
 dragstart, 195, 197  
 dropEffect, 201  
 dropover, 194  
 event.preventDefault(),  
 194  
 getData, 195  
 ikony przeciągania, 198  
 implementacja, 192  
 interakcje z danymi, 195  
 Internet Explorer, 194  
 OnDrop, 192  
 problemy z  
 ustawianiem  
 przeciąganych  
 danych, 197  
 procedury obsługi  
 zdarzeń, 192  
 przeciąganie danych do  
 innych aplikacji, 195  
 przeciąganie dowolnych  
 elementów, 198  
 setData, 195, 197  
 setDragImage, 198  
 strefa upuszczania, 192  
 przepływ, 70  
 przesłanianie domyślnych  
 opcji przeglądarek, 100  
 przesyłanie komunikatów,  
 213  
 przetwarzanie pikseli, 146  
 przewijanie filmu, 120  
 pubdate, 42  
 push, 227

## Q

querySelector, 133, 135,  
 146, 151

## R

radius, 136  
 ramki, 76  
 range, 88  
 rangeOverflow, 102  
 rangeUnderflow, 102  
 removeItem, 162  
 Renderowanie tekstu, 154  
 required, 92, 100  
 restore, 153  
 results, 89  
 retained-mode API, 140  
 role, 78, 97  
 role=banner, 65

role=main, 66  
 role=navigation, 64  
 rotate, 141  
 royalty-free, 115, 117  
 rozwinięcia, 161  
 ruby parentheses, 72  
 ruby text, 72  
 rysowanie, 131, 140, 219  
 gradient, 135  
 ścieżki, 138  
 wzorce, 135

## S

save, 153  
 search, 65, 89  
 Searchhi, 71  
 section, 52, 54  
 sectioning roots, 50  
 sekcje <aside>, 48  
 sekcje <nav>, 48  
 sekcje bez tytułu, 48  
 SELECT, 172  
 sessionStorage, 159, 164  
 setAttribute, 98, 128  
 setData, 195, 197  
 setDragImage, 198  
 setInterval, 142, 144, 218  
 setTimeout, 218  
 SharedWorker, 223  
 skanowanie wideo, 147  
 składnia XHTML, 18  
 składowanie danych, 157  
 opcje, 158  
 Web SQL Databases,  
 158, 166  
 Web Storage, 158  
 slogan, 49  
 specyfikacja ARIA, 67  
 specyfikacja HTML5, 12  
 specyfikacja W3C, 10  
 spellcheck, 79  
 spirala, 142  
 SQLite, 167  
 SQLTransaction, 175  
 src, 113  
 stan rysowania, 153  
 starsze przeglądarki, 29  
 start(), 123  
 step, 94  
 stepMismatch, 102  
 sterowanie multimediami,  
 118  
 stopka, 35, 43  
 <aside>, 35  
 <blockquote>, 44  
 wiele stopek w tym  
 samym elemencie, 43  
 stosowanie stylów kodu  
 HTML5, 29

strefa upuszczania, 192  
 strftime, 172  
 strict markup, 9  
 stroke, 139  
 strokeRect, 134  
 strokeStyle, 134, 139  
 struktura dokumentu, 17  
 struktura dokumentu ARIA, 65  
 struktura zasadniczych obszarów treści, 39  
 strumień danych, 226  
 styl stosowania wielkości znaków, 19  
 styl wypełnienia, 135  
 style CSS, 25  
 :heading(), 51  
 :invalid, 99  
 :required, 99  
 Internet Explorer, 27  
 nagłówki HTML5, 51  
 style dla nowych pól formularza, 99  
 subject, 78  
 suwaki, 88, 96  
 SVG, 10, 140  
 system GPS, 210

## Ś

ścieżki, 138

## T

tabindex, 79, 201  
 technika  
 przeciągnij i upuść, 191  
 technologia Comet, 226  
 tekst, 39  
 renderowanie, 154  
 tekst ruby, 72  
 tel, 89  
 tematyczna przerwa poziomu akapitu, 76  
 The Guardian, 58  
 obszar zasadniczej treści, 61  
 Theora, 114  
 time, 87  
 timers, 218  
 timestamp, 206  
 timeupdate, 127  
 TinyOgg, 117  
 toDataURL, 149, 150  
 tooLong, 102  
 transaction, 176  
 transakcje, 175  
 transformacje, 141  
 translate, 141, 142

treść strony, 36  
 tryb offline, 179  
 applicationCache, 180, 186  
 bufor, 189  
 CACHE, 183  
 CACHE MANIFEST, 181  
 FALLBACK, 182, 188  
 Firefox, 185  
 manifest, 179, 180  
 navigator.onLine, 188  
 NETWORK, 183  
 proces przeglądarka-serwer, 184  
 serwowanie pliku manifestu, 183  
 wykrywanie połączenia, 188  
 zerowanie bufora, 189  
 zmiany w lokalnym buforze, 186  
 tworzenie  
 baza danych, 168  
 gniazda, 228  
 struktura zasadniczych obszarów treści, 39  
 tabele, 169  
 wątki robocze, 217  
 type, 84  
 typeMismatch, 102

## U

umieszczanie klipu wideo, 108  
 update, 83  
 URL, 85  
 URL data, 149  
 ustawianie ciasteczka, 157  
 UTC, 42

## V

validity, 102  
 ValidityState, 102  
 valueMissing, 102  
 video, 109, 143  
 video.seekable, 123  
 video.seekable.end(), 124  
 VLC, 117  
 VP8, 114

## W

WAI-ARIA, 26, 63  
 łączenie ARIA z HTML5, 66  
 obszar zasadniczej treści, 66

poprawa dostępności, 97  
 role zasadniczych elementów, 65  
 role=banner, 65  
 role=main, 66  
 role=navigation, 64  
 struktura dokumentu, 65  
 zasoby, 66  
 walidacja kodu, 21  
 watchPosition, 204, 205, 206  
 wątek roboczy, 213, 216  
 debugowanie, 225  
 importowanie skryptów i bibliotek, 222  
 stosowanie, 217  
 tworzenie, 217  
 wątki robocze wewnątrz wątków roboczych, 220  
 wątki, 213  
 SharedWorker, 223  
 Web Accessibility Initiative's Accessible Rich Internet Applications, 63  
 Web Applications 1.0, 11, 12  
 Web Forms 2.0, 11  
 Web Hypertext Application Technology Working Group, 11  
 Web Messaging, 213  
 contentWindow, 214  
 czat, 214  
 event.origin, 215  
 event.source, 215  
 JSON, 215  
 postMessage, 214  
 wysyłanie danych, 215  
 wysyłanie komunikatów pomiędzy domenami, 214  
 Web Sockets, 10, 213, 226  
 API, 228  
 czat, 229  
 gniazda, 227  
 JSON, 229  
 opóźnienia, 227  
 tworzenie gniazda, 228  
 WebSocket, 228  
 Web SQL Databases, 158, 166  
 API, 167  
 executeSql, 168, 169, 170  
 forEach, 171  
 odpytywanie tabeli, 170  
 openDatabase, 168  
 otwieranie baz danych, 168  
 tabele, 169  
 transaction, 176

- Web SQL Databases
    - transakcje, 175
    - tworzenie baz danych, 168
    - tworzenie tabel, 169
    - wersje, 168
    - wstawianie danych, 170
    - wywołania zwrotne, 167
    - zapytania SQL, 169
  - Web Storage, 10, 158
    - API, 160
    - ciasteczka, 165
    - clear, 162
    - debugowanie, 164
    - dostęp do magazynu danych, 161
    - getItem, 160, 163
    - key, 162
    - localStorage, 159, 160
    - magazyny, 159
    - notacja JSON, 163
    - opcje zastępcze, 165
    - pobieranie danych, 160
    - removeItem, 162
    - rozwinęcia, 161
    - sessionStorage, 159, 160, 161
    - setItem, 160, 163
    - symulowanie magazynu localStorage, 165
    - typy danych, 163
    - ujednolicanie typów, 161
    - ustawianie danych, 160
    - usuwanie danych, 162
    - WebKit, 164
  - Web Workers, 213, 216
    - close, 218
    - debugowanie, 225
    - importowanie skryptów i bibliotek do wątku roboczego, 222
    - importScripts, 222
    - inicjowanie wątku roboczego, 219
    - licznik wątków, 221
    - onmessage, 218
    - operacje wewnątrz wątków roboczych, 218
    - postMessage, 217
    - przeglądarki, 217
    - przekazanie informacji do wątku roboczego, 217
    - setInterval, 218
    - setTimeout, 218
    - SharedWorker, 223
    - stosowanie wątków roboczych, 216
    - tworzenie wątków roboczych, 217
    - wątek roboczy, 216
    - wątki, 213
    - wątki robocze wewnątrz wątków roboczych, 220
    - współdzielenie obciążenia, 223
  - WebKit, 99, 164, 189
  - WebM, 114, 117
  - WebSocket, 228
  - week, 87
  - WHATWG, 10, 11
  - wideo, 107
    - cofanie, 125
    - kodeki, 113
    - kompresja, 116
    - obsługa w starszych przeglądarkach, 115
    - odtwarzanie, 120
    - pauza, 120
    - przesyłanie do urządzeń podręcznych, 116
    - przewijanie filmu, 120
    - przewijanie z podglądem, 125
    - szybkie przewijanie w przód, 125
  - width, 112
  - widziet w pływającej ramce, 214
  - wiele stopek w tym samym elemencie, 43
  - wielowątkowość, 216
  - willValidate, 102
  - window, 215
  - worker, 213
  - wprowadzanie danych, 84
  - ws://, 228
  - wstępne ładowanie metadanych, 125
  - wykorzystywanie wątków roboczych, 217
  - wyłączenie walidacji, 103
  - wymuszanie walidacji elementów, 102
  - wypełnianie, 91
  - wypychanie, 227
  - wrażenia regularne, 92
  - wyróżnianie słów, 69
  - wysyłanie komunikatów między domenami, 214
  - wzorce, 135
- ## X
- XHR, 13
  - XHTML, 9, 10
  - XHTML 2.0, 11
  - XHTML Strict, 11
  - XHTML Transitional, 11
  - XHTML5, 15
  - XML, 14, 15
  - XMLHttpRequest, 13, 218, 227
- ## Z
- zabezpieczenia dla ciasteczki, 159
  - zabezpieczenia przed kopiowaniem, 109
  - zajawka, 54, 58
  - zapewnienie zgodności wstecz ze starszymi przeglądarkami, 98
  - zapytania SQL, 169
  - zarządzanie wyłączeniem walidacji, 103
  - zastępowanie tła w wideo, 143
  - zdarzenia
    - loadeddata, 124, 144
    - loadedmetadata, 125
    - onforminput, 105
    - oninputchange, 104
    - onmessage, 218
    - timeupdate, 127
  - zestaw znaczników w wersji ścisłej, 9
  - znaczniki samozamykające, 18
  - zwijalny obszar, 68

WPROWADZENIE DO

# HTML5

HTML5 – początkowo po cichu, ostatnio coraz głośniejsz – zaczyna gościć w naszych przeglądarkach internetowych. Wbudowane mechanizmy obsługi multimediów, lepsze wykorzystanie formularzy, nowe struktury i semantyka, dzięki którym Twój serwis stanie się bardziej dostępny, to tylko część z nowinek ostatniej wersji języka HTML. Zastanawiasz się, jak wykorzystać go na Twoich stronach?

Ta książka odpowie na to i wiele innych pytań! Kompletnie źródło informacji na temat HTML5 jest w zasięgu Twojej ręki. W trakcie lektury poznasz nowe struktury i ich przeznaczenie oraz zmodyfikowane znaczniki z wcześniejszych wersji. W kolejnym rozdziale zaznajomisz się szczegółowo z nowymi możliwościami formularzy oraz technikami ich tworzenia. Projektowanie formularzy w nowej odsłonie tego języka to naprawdę czysta przyjemność! Ponadto zobaczysz, jak szybko i łatwo można dodać do strony elementy wideo i audio. Dodatkowo nauczysz się wykorzystywać mechanizmy do przechowywania danych oraz geolokalizacji. Dzięki tej książce poczujesz powiew świeżości w świecie stron internetowych!

- ▼ Struktura dokumentu HTML
- ▼ Struktury zasadniczych obszarów treści
- ▼ Wykorzystanie konspektu
- ▼ WAI-ARIA
- ▼ Wykorzystanie globalnych atrybutów
- ▼ Elementy do wprowadzania danych w formularzach
- ▼ Natywna obsługa multimediów
- ▼ Rozwiązywanie problemów z kodekami
- ▼ Obsługa wideo w starszych przeglądarkach
- ▼ Właściwości elementu Canvas
- ▼ Składowanie danych
- ▼ Metody dostępu do magazynu danych
- ▼ Praca w trybie offline
- ▼ Wykorzystanie techniki „przeciągnij i upuść”
- ▼ Geolokalizacja – możliwości oraz sposoby wykorzystania

## Twórz najnowocześniejsze witryny WWW!



**Helion**

Sprawdź najnowsze promocje

🔗 <http://helion.pl/promocje>

Książki najchętniej czytane

🔗 <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

🔗 <http://helion.pl/nowosci>

**Helion SA**

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 230 98 83

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

<http://helion.pl>

Nr katalogowy: **6183**



Księgarnia internetowa:

<http://helion.pl>



Zamówienia telefoniczne:

**0 801 339900**



**0 601 339900**

**helion.pl**  
księgarnia  
internetowa

Cena 39,00 zł

ISBN 978-83-246-3045-5



9 788324 630455

Informatyka w najlepszym wydaniu